

**Amendment to the Drawings:**

The Examiner objected to the drawings under 37 C.F.R. 1.83(a) asserting that the drawings fail to show every feature of the invention specified in the claims. Specifically, the Examiner asserts that certain elements of claims 1, 11, 17, 22, and 24 are not shown in the drawings. Based on at least the following grounds, Applicants respectfully traverse.

The elements of Claims 1, 11, 17, 22 and 24 in question are shown in, at least, one or more of Figures 2-4.

Specifically, shown in Figure 2 are the components of one embodiment of the present application that are provided by developers, including the specification of separate Applications, Data Models and Integration Components. As shown in Figure 2, the models are a separate component and, thus, obviously decoupled.

In Figure 3, one embodiment of the deployment of models (316) is referenced. As shown, this deployment is separate from other components (314), thus illustrating their decoupled nature. Further, in Figure 3, the deployment of Data Model components (316) separately from Application components (314), is shown, again providing an environment for independent changes (i.e., making alterations to the model without any changes to other components).

In Figure 4, the update of data models as a separate step (412) from the other process steps, including the enhancement of device components (414) and integration components (416) is shown according to one embodiment. Each of these steps in the process were drawn separately at least to show how they are decoupled from one another and how each part of the process can be performed independently. In other words, the diagram shows that if the Mobile Data Model is updated, the Device Component enhancements might then be done (if needed), then Integration Components enhancements might be done (if needed). However, for the present purpose, the diagram shows how these steps are separate decision steps in the process. Similarly, enhancements to the Device Components and Integration Components could be done in cases where the Mobile Data Model had not been updated. Again, all of these are separate, independent processes. As such, the present specification provides, in one embodiment, an environment that clearly shows how data model

changes can be made without significant supplementary programming (i.e., enhancements to other components).

Figure 4 also shows at least a portion of the present invention's usage of rules. Specifically, the patent specification explains how "transactions may be applied directly to the data store, then sent to the rules processing engine to be distributed to consumers in the mobile domain that might be 'interested', at 430". Continuing, the specification provides, in one embodiment, that "[a]t step 430, once data transactions are applied to server based data stores, they may be additionally processed by the server based mobile computing system in order to determine if any consumers within the mobile domain might also need to be informed about the data operations contained therein. This processing is handled by a special rules engine within the server based mobile computing system. This rules engine may be driven by special conditional logic statements developed by the administrator of the system." Such conditional logic statements are described, in one embodiment, where it is mentioned that "[s]pecial conditional logic statements which drive the rules engine in the server based mobile computing system may be created by the system administrator using the graphical user interface provided by the mobile computing system, at step 427. These rules may control how data that is applied to the server based mobile domain is distributed to other consumers in the domain." Thus, Figure 4 has illustrated the operation and use of a particular embodiment of mobile domain that may be incorporated into a distributable software platform.

Further, going back to the provisional patent application filing to which the present application claims priority, decoupled data models are also shown in the illustration. For example, Page 20 (from 20-Jan-1999) of the provisional filing, shows a Data Model as a separate, decoupled component (note the "boomerang" shaped data model in the lower-left corner). Another reference can be found on Page 27 (from 08-Feb-1999) of the provisional filing, where it is shown that the data model is the independent link between an Application and a Data Store (the storage place for data).

### **REMARKS**

Claims 1-26 were rejected by the Examiner. Claims 1-26 are still pending. Reconsideration is respectfully requested in view of the amendments above and the following remarks.

#### **Claim Rejections under 35 U.S.C. § 112**

Claims 11-21 were rejected under 35 U.S.C. § 112, 1<sup>st</sup> ¶, as failing to comply with the written description requirement. Specifically, the Examiner asserted the originally filed specification does not provide support for certain limitations – "... wherein the mobile data model is ... operable to enable changes in data structure and data handling without requiring programmatic changes in the enterprise back end ..." and "... changes to the mobile data model effect changes in system data descriptions and rules governing data handling without requiring programmatic changes in application included in an enterprise back-end or mobile device" – in Claims 11 and 17, respectively. Applicants respectfully traverse based on at least those arguments presented above with respect to the Drawings and further on the following grounds.

In making his rejection of Claim 11, the Examiner specifically points out how on page 16, line 16 - page 17, line 3 one embodiment of the specification states that the data model is built before the server or integration components and that if requirements change, the integration components may be changed. The Examiner asserts that this line shows how it is the integration components enable the change, not the model. Applicants respectfully assert that the Examiner has chosen to view these teachings out of context.

Specifically, according to one embodiment of the present application, a mobile data model is built (and updated) before the integration components can be built (or updated). However, just because one embodiment provides for the option that integration components may change as needed, does not mean that such an embodiment is the only way that change can be introduced into the system. For example, on page 16, lines 14-15, one embodiment of the present application states that "[a]s application requirements change, the developer may return to the mobile data model to update it as needed," a direct analogue to the statement the Examiner is keying off of on page 17, lines 2-3

where we say that "[a]s application requirements change, the developer may enhance the integration components as needed." Accordingly, either update may enable change.

Further, in various places in the present specification, a discussion is provided regarding how "portions" of the data model may be instantiated on the device or to access the backend. As an example, see page 30, lines 7-8 where it is stated that "[w]hen a mobile user synchronizes and colonizes a hand-held device, at least a portion of the mobile data model may be instantiated on the device." The reason the term "portion" is used, according to one embodiment, is to point out how it's possible that only certain segments of the model may be interesting to particular clients (and the corresponding mobile applications which are also deployed there) or backend connections (and the integration components which are deployed to handle that side of things). A key point here is that it should be fairly obvious that if someone changed those portions of the model that hadn't been deployed to that particular client or integration, that that code wouldn't have to change. In reality, that code probably wouldn't even realize that there was a change made.

In addition, this point is again made, albeit from a different direction. The present specification points out how a significant role of a mobile data model is to provide abstraction between components in the system. For example, on page 34, lines 18-20, we say exactly that: "In effect, the mobile data model may provide a layer of abstraction between a back-end database and a mobile application." A key value of abstraction is to isolate components from changes to other components. Thus a key purpose of the data model is to allow changes without changing the other components in the system, either programmatically or any other way. The specification continues by giving an obvious scenario for how changes could be made without the components being affected. On page 34, line 20 - page 35 line 2, we explain how "an integration component may access a domain data store instantiated from a mobile data model or a portion thereof, and a mobile applications may access a mobile data store instantiated from the same mobile data model or a portion thereof on an individual hand-held device.". One point of that example is to illustrate how different components in the system may actually be referencing different portions of the larger data model. Thus, it should be obvious that someone could change a particular portion of the model that hadn't been deployed to the mobile data store (or referenced by the mobile application) or change a portion of the model that hadn't been deployed to the domain data store (or referenced by the

integration component). In either of those cases, the change to the data model would not require changes in these components. In fact, those components wouldn't even be aware that the change occurred.

In still another example, the specification discusses how new classes can be added to a model. These changes could then be deployed to only new users that had been added to a solution. For example, see page 37, line 19 - page 38, line 2 where the specification states "[a]s the use of the mobile domain solution matures, modifications to the solution may be warranted. For example, an enterprise may elect to include new types of mobile employees and new classes might have to be added to the existing model to support these new user types. Because the mobile data model can represent the underpinnings of a unifying schema, the model may be re-deployed to all users or just to those new users added to the solution." One key point here is how we allow for the selective deployment of changes. In this particular case, the model may not be deployed to existing users, which obviously would obviate the need for any programmatic changes.

In addition to examples that wouldn't require any programming, it's an obvious conclusion that the present application could also support changes that certain components might want to incorporate into their programming, along with components that wished to ignore the changes. The key, therefore, is that the teachings of the present specification allow for the possibility that components may incorporate changes to the model, but they don't have to.

For at least the foregoing reasons, Applicants respectfully request that the Examiner reconsider the rejection of Claim 11 under 35 U.S.C. § 112, 1<sup>st</sup> ¶, withdraw the rejections and allow Claim 11.

Claims 12-16 depend from and provide further patentable limitations to Claim 11. Accordingly, Applicants respectfully request that the Examiner reconsider the rejection of Claims 12-16, withdraw the rejections and allow Claims 12-16.

In rejecting Claim 17, the Examiner argues that according to the passage on page 34, line 14 - page 35, line 2 that our data model is not "operable", because the data model is instantiated as a data store which is what the integration component or mobile application interfaces with.

Furthermore, the Examiner argues that any change in the data model would require a new instantiation of the data store, and finally that any change in the data model would require modifications to associated integration components or applications. In general, the examiner argues that we are "silent" in regards to our ability to change the data model without requiring programmatic changes. Applicants respectfully traverse based on at least those arguments presented above with respect to the Examiner's challenges to the Drawings and Claim 11 and further based on the following grounds.

According to the teachings of the present application, neither the mobile applications nor the integration components ever access the data store directly. Instead, they always access a data store in the context of the data model. Without the data model, they wouldn't understand the structure of the data store. Also, it is not true that changes to the data model require a new instantiation of the data store. Rather, as we state on page 24, lines 11-12, "[i]f a data store deployment is received for an already deployed data store, the updated data model may be used to alter the structure of the database (as required)". Note the emphasis on altering the structure (versus recreating the whole data store).

Further, the Examiner asserts that while the present specification does state how "rules in the system dictate", on page 13, lines 19-21 and page 13, lines 1-4, our reference to "rules governing data handling" was not. Applicants respectfully traverse noting that the Examiner's assertion is misplaced.

Specifically, on at least pages 24 and 25 of the specification, a detailed rundown of how rules work is provided. For example, on page 24, lines 6-8, one embodiment of the present specification explains how "transactions may be applied directly to the data store, then sent to the rules processing engine to be distributed to consumers in the mobile domain that might be 'interested,' at 430". Later, on page 24, line 18 - page 25, line 2, the present specification describes how "[a]t step 430, once data transactions are applied to server based data stores, they may be additionally processed by the server based mobile computing system in order to determine if any consumers within the mobile domain might also need to be informed about the data operations contained therein. This processing is handled by a special rules engine within the server based mobile computing system. This rules

engine may be driven by special conditional logic statements developed by the administrator of the system." These conditional logic statements are described, according to one embodiment, on page 25, lines 13-18 where it is provided that "[s]pecial conditional logic statements which drive the rules engine in the server based mobile computing system may be created by the system administrator using the graphical user interface provided by the mobile computing system, at step 427. These rules may control how data that is applied to the server based mobile domain is distributed to other consumers in the domain. Thus, Figure 4 has illustrated the operation and use of a particular embodiment of mobile domain that may be incorporated into a distributable software platform."

Further, the Examiner argues that while on page 24, line 7 we refer to a "rules processing engine" that we do not mention any changes in the rules engine in direct response to a change in the mobile data model. Applicants respectfully traverse based on at least the arguments presented above and further based on the following grounds.

According to Applicants' present application, rules control the flow of data transactions which are described by the mobile data model. In addition, the conditional logic statements which are the heart of rules are formulated based on the descriptions found in the model. As such, according to one embodiment, there are both direct and indirect impacts to rules when the data model changes.

Claims 18-21 depend from and provide further patentable limitations to Claim 17. Accordingly, Applicants respectfully request that the Examiner reconsider the rejection of Claims 18-21, withdraw the rejections and allow Claims 18-21.

Claims 11-21 were rejected under 35 U.S.C. § 112, 2<sup>nd</sup> ¶, as failing to set forth the subject matter which the applicants regard as their invention. Specifically, the Examiner alleges that statements made in Applicants' response to office action filed 09.15.2005 evidence that claims 11-21 fail to correspond in scope with that which Applicants regard as the invention. Based on at least those arguments presented above with respect to the Examiner's challenges to the Drawings and Claims 11 and 17, Applicants respectfully traverse.

For at least the foregoing reasons, Applicants respectfully request that the Examiner reconsider the rejection of Claims 11-21 under 35 U.S.C. § 112, 2<sup>nd</sup> ¶, withdraw the rejections and allow Claims 11-21.

**Claim Rejections under 35 U.S.C. § 103(a)**

Claims 1-26 were rejected under 35 U.S.C. § 103(a) as being unpatentable over prior art of record U.S. Patent No. 5,857,201 to Wright et al. (Wright) in view of U.S. Patent No. 5,295,222 to Wadhwa et al. (Wadhwa) in view of U.S. Patent No. 6,332,163 to Bowman-Amuah (Bowman-Amuah).

**ARGUMENT**

- I. Applicants respectfully assert that the 35 U.S.C. § 103 rejection is improper on the grounds both that, contrary to the Examiner's arguments, there is no motivation to combine the teachings of Wright with those of Wadhwa, and that Wright teaches away from its combination with Wadhwa.**

The Examiner alleges that "it would have been obvious to one of ordinary skill in the art at the time the invention was made to use Wadhwa's data model including data relationship attributes with Wright's software platform. One of ordinary skill would have been motivated to provide a data model that can be readily re-used in subsequent applications. (Office Action mailed 11.02.2005.)

In view of the arguments set forth below, Applicants respectfully assert that the Examiner's arguments are misplaced on at least the grounds that the Examiner's cited motivation for combining Wadhwa with Wright – "re-use" – exists in Wright and, therefore, no motivation as asserted by the Examiner may be found for their combination. In addition, the teachings of Wright contradict the teachings of Wadhwa thereby prohibiting their combination as a basis for a valid rejection under 35 U.S.C. § 103.

- a. Wright teaches away from Wadhwa in that Wright teaches platform independent applications development and coding whereas Wadhwa is directed to resolving issues surround the development of code targeted to many specific hardware platforms.**

If a first prior art reference teaches away from a second prior art reference, that finding alone can defeat an obviousness claim based on a combination of the two references. *See, Winner Int'l Royalty Corp v. Wang*, 202 F.3d 1340 (Fed. Cir. 2000). A reference may be said to teach away when a person of ordinary skill, upon reading the first reference, would be led in a direction divergent from the path taken in the second reference. *See, In re Haruna*, 249 F.3d 1327 (Fed. Cir. 2001).

Wright discloses a FormLogic engine allowing applications to execute on a variety of platforms. (See, e.g., Wright at 5:16-29.) As expressly stated by Wright, the FormLogic engine, the engine upon which the applications of Wright are designed to operate, "is a hardware independent virtual machine that allows a single application to work on various hardware platforms." (Wright at 5:25-26.)

Applicants agree with the Examiner, Wadhwa generally teaches object oriented programming. Specifically, Wadhwa discloses computer-aided software engineering facilities for assisting programmers or system developers in the design, development and testing of computer programs for use in multiprocessor systems. (See, e.g., Wadhwa at 1:16-19.) According to Wadhwa, in order to "design an application using the CASE facility, a developer must decompose the application into specified logical parts, and assemble them into a program." (Wadhwa at 6:39-42.) "The different parts of an application are expressed as entities and are linked by relationships." (Wadhwa at 6:44-46.) "With the Entity-Relationship model in place program construction can begin." (Wadhwa at 11:21-22.)

In Wadhwa, entity-relationship models are employed to simplify the generation of code for an application intended for use on variety of hardware platforms. (See, e.g., Wadhwa at 1:16-19.) In other words, Wadhwa describes a system wherein a model of a desired application is developed. Then, once an application is modeled, Wadhwa provides CASE facilities which are configured to generate code for each of the targeted hardware platforms on which the application is intended to execute. To be clear, Wadhwa is directed to providing, potentially, numerous iterations of code for a single application such that the application may be executed on a variety of hardware platforms. It is in this regard that Wadhwa teaches the use of a reusable model – for the sole purpose of generating,

for a single application, numerous versions of code such that the single application may be executed on a variety of hardware platforms.

From the indisputable teachings of both Wright and Wadhwa, it is clear that the implementations in each are in conflict – Wadhwa teaching platform dependence and, in stark contrast, Wright teaching platform independence. On this basis alone, the combination of Wadhwa and Wright should be rejected as this basic distinction – platform independent versus platform dependent code development – would indisputably lead any person of ordinary skill in the art reading Wright down a path entirely “divergent from the path taken in Wadhwa”. *See, In re Haruna*.

In fact, Wright should be considered to entirely supplant the teachings of Wadhwa. Based on the express teachings of Wright, it would be evident to one of ordinary skill in the art that to deploy an application in the Wright model one would need to develop and code only one version of the desired application. Thus, Wright clearly obviates any need for the entity-relationship model of Wadhwa through the use of Wright’s “hardware independent virtual machine that allows a single application to work on various hardware platforms”. Again, a person of ordinary skill in the art reading Wright would be led on a path – platform independent code generation – entirely divergent from the path taught by Wadhwa – platform dependent code generation. *Id*.

Accordingly, Applicants respectfully assert that the teachings of Wright are in conflict with the teachings of Wadhwa and, further, the Wright disclosure teaches away from its combination with Wadhwa. There is no suggestion to combine if a first reference teaches away from its combination with a second reference. *See, Tec Air, Inc. v. Denso Mfg. Michigan Inc.*, 192 F.3d 1353 (Fed. Cir. 1999). Therefore, Applicants respectfully request that the Examiner reconsider the combination of the Wright and Wadhwa references, withdraw the rejections based on their combination and allow Claims 1-26 rejected thereunder.

- b. The Examiner’s cited motivation for the combination of Wadhwa with Wright – the motivation of “re-use” – should be rejected as Wright expressly teaches re-use in the form “a hardware independent virtual machine that allows a single application to work on various hardware platforms.”**

A showing of obviousness requires a motivation or suggestion to combine or modify prior art references, coupled with a reasonable expectation of success. *See, Boehringer Ingelheim Vetmedica, Inc. v. Schering Plough Corp.*, 320 F.3d 1339 (Fed. Cir. 2003). A showing of a suggestion, teaching, or motivation to combine must be clear and particular; broad conclusory statements regarding the teaching of multiple references, standing alone, are not evidence. *See, In re Dembiczak*, 175 F.3d 994 (Fed. Cir. 1999). The test for establishing an implicit teaching, motivation, or suggestion is what the combination of statements of one or more prior art references would have suggested to one of ordinary skill in the art. *See, In re Kotzab*, 208 F.3d 1352 (Fed. Cir. 2000). Importantly, these statements must be considered in the context of the teachings of the entire reference. *Id.* There is no suggestion to combine if a first reference teaches away from its combination with a second reference. *See, Tec Air, Inc. v. Denso Mfg. Michigan Inc.*

In addition to those reasons stated above and believed to justify the withdrawal of the claim rejections based on the combination of Wright and Wadhwa, Applicants respectfully assert that the rejections based on the combination of Wright and Wadhwa should also be withdrawn on the grounds that there is no motivation for their combination for at least the following reasons.

Applicants and the Examiner agree, "Wright does not expressly teach a data model". (Office Action mailed 11.02.2005, p. 9.) As such, Applicants respectfully assert that there can be no motivation for one of ordinary skill in the art merely to replace the implied model of Wright with the "model" of Wadhwa. Therefore, in order to properly combine the teachings of the Wright and Wadhwa references, there must be some alternative motivation. Thus, according to the Examiner and as mentioned above, the impetus for the combination of the Wright and Wadhwa references purportedly lies in the motivation of one of ordinary skill in the art to provide Wright with "a data model that can be readily re-used in subsequent applications."

As stated above, Wright discloses a FormLogic engine allowing applications to execute on a variety of platforms. (See, e.g., Wright at 5:16-29.) As expressly stated by Wright, the FormLogic engine, the engine upon which the applications of Wright are designed to operate, "is a hardware independent virtual machine that allows a single application to work on various hardware platforms." (Wright at 5:25-26.) With this statement, Wright unquestionably contemplates and

provides for the extensive reusability of applications developed and coded for use therewith. Therefore, a person of ordinary skill in the art reading Wright **would not** be motivated to provide Wright “with a data model that can be readily re-used in subsequent applications” because one of ordinary skill in the art reading Wright would easily and immediately recognize that Wright already provides for the reusability of applications through Wright’s express teaching of the use of platform independent application development and coding tools. Someone familiar with Wright would not be motivated to add Wadhwa teachings because that would be a step backwards.

In view of the above arguments, Wright clearly teaches away from its combination with Wadhwa – Wright seeks to develop and code a single application while Wadhwa seeks to generate application code for each and every different hardware platform upon which the desired application is intended to execute. Thus, as Wright already teaches reusability, one of ordinary skill in the art would not be motivated on those same grounds to seek out utilize the incompatible teachings of Wadhwa.

Accordingly, Applicants respectfully assert that the teachings of Wright are in conflict with the teachings of Wadhwa and, further, the Wright disclosure teaches away from its combination with Wadhwa. There is no suggestion to combine if a first reference teaches away from its combination with a second reference. *See, Tec Air, Inc. v. Denso Mfg. Michigan Inc.*, 192 F.3d 1353 (Fed. Cir. 1999). Therefore, Applicants respectfully request that the Examiner reconsider the combination of the Wright and Wadhwa references, withdraw the rejections based on their combination and allow Claims 1-26 rejected thereunder.

**II. Applicants respectfully assert that the 35 U.S.C. § 103 rejection is improper on at least the grounds both that, contrary to the Examiner’s arguments, there is no motivation to combine the teachings of Wadhwa with those of Bowman-Amuah and that Bowman-Amuah teaches away from its combination with Wadhwa.**

A showing of obviousness requires a motivation or suggestion to combine or modify prior art references, coupled with a reasonable expectation of success. *See, Boehringer Ingelheim Vetmedica, Inc. v. Schering Plough Corp.*, 320 F.3d 1339 (Fed. Cir. 2003). A showing of a suggestion, teaching, or motivation to combine must be clear and particular; broad conclusory statements regarding the teaching of multiple references, standing alone, are not evidence. *See, In re*

*Dembiczak*, 175 F.3d 994 (Fed. Cir. 1999). The test for establishing an implicit teaching, motivation, or suggestion is what the combination of statements of one or more prior art references would have suggested to one of ordinary skill in the art. *See, In re Kotzab*, 208 F.3d 1352 (Fed. Cir. 2000). Importantly, these statements must be considered in the context of the teachings of the entire reference. *Id.*

If a first prior art reference teaches away from a second prior art reference, that finding alone can defeat an obviousness claim based on a combination of the two references. *See, Winner Int'l Royalty Corp v. Wang*, 202 F.3d 1340 (Fed. Cir. 2000). There is no suggestion to combine if a first reference teaches away from its combination with a second reference. *See, Tec Air, Inc. v. Denso Mfg. Michigan Inc.* A reference may be said to teach away when a person of ordinary skill, upon reading the first reference would be led in a direction divergent from the path taken in the second reference. *See, In re Haruna*, 249 F.3d 1327 (Fed. Cir. 2001).

- a. **Mere recitation in the same sentence of a “de-coupled” and “data models” does not support the Examiner’s assertion that Bowman-Amuah “teaches that it is important to decouple a data model.”**

According to the Examiner, Bowman-Amuah “teaches that it is important to decouple a data model” at 236:51-56. This assertion simply is not supported in the language cited.

A reading of the cited portion of Bowman-Amuah makes it clear that Bowman-Amuah is not discussing a decoupled data model but is instead teaching “de-coupled communication” in systems supporting “volatile and constantly changing object models, data models and data structures.” (236:51-55.) To overcome the difficulties presented in such systems, Bowman-Amuah teaches the use of a “shared format”. (236:41-43.) In the words of Bowman-Amuah, this shared format overcomes the difficulties presented in systems supporting “volatile and constantly changing object models, data models and data structures” by acting as “a secret decoder ring for systems sending and receiving messages.” (236:45-47.) To facilitate these desired “de-coupled” communications, Bowman-Amuah’s “shared format” will allow “systems to convert structured data (objects, strings, etc.) into raw data and raw data back into structured data.” (236:47-49.) As such, Bowman-Amuah

teaches, at best, the provision of nothing more than a translator between end points in a communication system.

The Examiner acknowledges his awareness of Bowman-Amuah's "shared format" in stating that "one of ordinary skill in the art would have been motivated to use a shared format that decouples the model in order to efficiently handle changes to the system." (emphasis added) (Office Action mailed 11.02.2005, p. 10.) However, what the Examiner seemingly fails to appreciate is Bowman-Amuah's acknowledgement that its data model is separate and apart from the "shared format" the Examiner relies upon noting that "every change to the object model, data model of data structure causes a reimplementation of the 'shared format.'" As such, the Examiner's citation to Bowman-Amuah's "de-coupled communications" refers to the "shared format" Bowman-Amuah seeks to employ as a translator between communication endpoints and not to any data model to which Bowman Amuah might refer.

In view of the foregoing, Bowman-Amuah fails to teach a decoupled data model as described and claimed in the present application. What's more, the Examiner's assertion that "it would have been obvious to one of ordinary skill in the art at the time of the invention was made to use Bowman-Amuah's teachings of a decoupled data model with Wadhwa's independent data model" is misplaced at least on the grounds that Bowman-Amuah fails to teach a decoupled data model. Accordingly, Applicants respectfully request that the Examiner reconsider the rejection of Claims 1-26 based on the combination including Bowman-Amuah, withdraw the rejection and allow Claims 1-26.

- b. Bowman-Amuah teaches away from its combination with Wadhwa at least on the grounds that the very problems Bowman-Amuah seeks to avoid through its teachings of "de-coupled communications" is an expressly acknowledged drawback of Wadhwa's teachings.**

The Examiner asserts that "one of ordinary skill in the art would have been motivated to use a shared format that decouples the model in order to efficiently handle changes to the system." (Office Action mailed 11.02.2005, p. 10.) In explaining the motivation for its shared format, Bowman-Amuah states that "in a constantly changing system, a statically defined 'shared format' doesn't work very well. Every change to the object model, data model of data structure causes a

**reimplementation of the 'shared format.' Each reimplementation results in a redesign, recompile, and retest of the changed code.** (236:57-62.)

In indisputable contrast to the desired system of Bowman-Amuah, Wadhwa acknowledges a significant shortcoming of its teachings based on an entity-relationship model. Specifically, Wadhwa states that **"small changes in this system can have large consequences. In general, any entity type that owns, uses or includes entities that have been changed will have to be reprepared or modified."** (17:1-4). As such, the system of Wadhwa where "any entity type that owns, uses or includes entities that have been changed will have to be reprepared or modified" is indisputably disavowed by Bowman-Amuah as an undesirable implementation.

In view of the foregoing arguments, it is clear that a person of ordinary skill in the art would be lead down a path divergent from the teachings of Wadhwa. *See, In re Haruna*. That Bowman-Amuah teaches away from its combination with Wadhwa precludes Wadhwa or Bowman-Amuah from including a motivation for their combination. *See, Tec Air, Inc. v. Denso Mfg. Michigan Inc.* Accordingly, the combination of Wadhwa and Bowman-Amuah as a basis for rejecting claims 1-26 of the present application is improper. Therefore, Applicants respectfully request that the Examiner reconsider the rejection of Claims 1-26 based on a combination of references including Bowman-Amuah and Wadhwa, withdraw the rejection and allow Claims 1-26.

**III. Even if inappropriately permitted, the combination of Wright and Wadhwa fails to disclose, teach or otherwise suggest a data model as claimed by Applicants.**

- a. The Examiner and Applicants agree, Wright DOES NOT "expressly disclose a data model defining one or more data element, data relationship, data dependency and data distribution attributes".**

The Examiner and Applicants agree, Wright "does not expressly disclose a data model defining one or more data element, data relationship, data dependency and data distribution attributes". (Office Action mailed 11.02.2005, p. 9). In contrast, Wright et al. detail an application based, programmatic solution directed to improving the capability of custom designed programs.

- b. Wadhwa, through its usage of an "entity-relationship model", discloses a model representing only the association (relationship) of data (entities).**

The Examiner has stated that “Wadhwa teaches that a data model that defines at least data relationships can be used for generation and distribution of applications.” (Office Action mailed 11.02.2005, p. 9). Nowhere in Wadhwa is an entity-relationship model defined to include “data dependency and data distribution attributes” as claimed by Applicants.

Specifically, Wadhwa, in its description of a model used to generate platform dependent application code, states that the “different parts of an application are expressed as entities and are linked by relationships.” (6:44-46). Continuing, Wadhwa states that “an entity is something real or abstract about which information is recorded.” (6:47-48). A relationship, according to Wadhwa, is an “association between entities” and is “defined by attributes”. (6:59-63). Wadhwa never mentions a data model defining data dependencies or data distribution attributes in its discussion of an entity-relationship model – as will be noted in greater detail below, Wadhwa has absolutely no use for such attributes. Nowhere does Wadhwa discuss data dependency or data distribution attributes, let alone a data model defining data element, data relationship, data dependency and data distribution attributes required for interfacing a mobile software application with a backend application.

**c. Directed to hardware platform specific application code generation in a multiprocessing or distributed processing environment, there is no basis in Wadhwa for the use of data distribution and data dependency attributes.**

- i. The present application contemplates asynchronous communication between periodically connected mobile computing systems and devices whereas Wadhwa contemplates persistently connected processing nodes.

In the present application, a data model defining data element, data relationship, data dependency and data distribution attributes required for interfacing a mobile software application with a backend application is provided. In this vein, the present application contemplates the need to share data by and between multiple disconnected clients as well as to communicate asynchronously between the periodically disconnected clients and one or more backend applications. In such an environment, managing data dependencies and distribution is critical.

As a solution for generating application code in a multiprocessor or distributed processing environment, Wadhwa contemplates an environment of persistent network connectivity –

multiprocessing and distributed processing systems typically being hardwired together over one or more network connections. As such, the system of Wadhwa may be considered a single entity consisting of each connected device participating in processing, thereby, all communications between the devices may be considered internal communication within the distributed processing configuration.

ii. Multiprocessor and distributed processing solutions utilize highly structured applications with extensive process order management.

As suggested by the manner in which the present application describes and claims its data model, the data model of the present application plays an important role in interfacing a mobile software application with a backend application. This too provides a context in which it can be seen that the data model of the present application is not disclosed, taught or otherwise suggested by Wadhwa.

In contrast to the environment in which the present application is designed to operate, Wadhwa is directed to operation in a highly structured and managed operating environment. By their very nature, distributed or multiprocessor applications – whose fundamental difference with conventional, single node applications is that processing operations are shared among nodes to increase processing power and decrease processing time – direct which portion of what processing needs to occur when, in accordance with what priority, which portions of processing may be performed in parallel, etc. Such an environment is distinct from a remote or mobile computing system at least in the aspect that random connections, concurrency, and other issues that accompany the remote computing system are absent.

iii. Wadhwa is concerned only with the distribution of software generated from the entity-relationship model, performed by a Software Distribution System independent of the entity-relationship model.

Wadhwa discloses a “Software Distribution System which automates and controls migration of an application. The system manages the release of software to targeted computers. The Software Distribution System solves the problem of synchronizing distribution of software located, for example, on hundreds of personal computers.” (6:24-29). As such, distribution in Wadhwa – that is

distribution related to the dissemination of a distributed or multiprocessor application to each of its targeted nodes – is orchestrated by a Software Distribution System entirely separate and apart from Wadhwa's "Entity-Relationship model". Consequently, Wadhwa lacks any motivation to incorporate attributes concerning "software distribution," let alone the data dependency and distribution attributes contemplated by the present application, into its purported data model. In fact, Wadhwa's specific inclusion of a Software Distribution System teaches away from the incorporation of any aspect of distribution into its Entity-Relationship model.

- d. In contrast to Wadhwa's Entity-Relationship model, the data model of the present application defines and is actively employed, accessed or otherwise leveraged in interfacing a mobile software application with a backend application.**

As described above, Wadhwa develops an Entity-Relationship model from which hardware environment or platform specific application code is generated. "The entity –relationship model is not the actual program." (7:24-25). Subsequent to code generation, the entity-relationship model from which application code was developed is stored for possible later re-use or otherwise discarded. (See, e.g., 11:59-60). "When the modules of an application have been successfully prepared, they are ready to be transported to their target environments," *i.e.*, the application modules and not the entity-relationship model are transported to their target environments. (15:42-44).

In contrast, the present application describes and claims "a data model defining data elements, data relationships, data dependencies and data distribution attributes required for and actively employed in interfacing a mobile software application with at least one of the plurality of backend applications". Unlike, Wadhwa, an embodiment of the present application anticipates leveraging the data model it provides during interfacing between a mobile software application and a backend application.

In view of the foregoing arguments, Applicants respectfully submit that the rejection of Claims 1-26 under 35 U.S.C. § 103(a) as being unpatentable over Wright in view of Wadhwa and further in view of Bowman-Amuah is improper as the references may not be properly combined to form a basis for rejection. As it is improper to combine the teachings of Wadhwa with Wright and the teachings of Bowman-Amuah with Wadhwa, Wright is the only reference that remains.

Application No. 09/848,770  
Amendment dated May 2, 2006  
Reply to Office Action of November 2, 2005

Regarding Wright, the Examiner has already stated that the claims presented herein are not disclosed in Wright standing alone. (Office Action mailed 11.02.2005, p. 2.) Thus, in view of the Examiner's comments and the improper combination of Wadhwa and Bowman-Amuah to the teachings of Wright, Applicants respectfully request that the Examiner reconsider the rejection of Claims 1-26, withdraw the rejection and allow Claims 1-26.

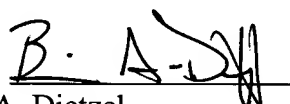
### CONCLUSION

In light of the remarks set forth above, Applicants believe that they are entitled to a letters patent in the present matter. Applicants respectfully solicit the Examiner to expedite prosecution of this patent application to issuance. Should the Examiner have any questions or feel that further prosecution of this matter may be expedited through an interview, the Examiner is encouraged to telephone the undersigned.

The Commissioner is authorized to charge any additional fees which may be required, including petition fees (Docket No. 26625.704) and extension of time fees (Docket No. 1000.060), to Deposit Account No. 23-2415.

Respectfully submitted,

Date: May 2, 2006

By:   
Brian A. Dietzel  
Registration No. 44,656

WILSON SONSINI GOODRICH & ROSATI  
650 Page Mill Road  
Palo Alto, CA 94304-1050  
(512) 338-5423  
Client No. 021971